Attorney Docket No. GCENP006

PATENT APPLICATION

TECHNIQUES FOR PROVIDING INTEROPERABILITY AS A SERVICE

Inventors: Ron Palmeri of
Berkeley, California
United States citizen

Byrne Reese of
Berkeley, California
United States citizen

Assignee: Grand Central Communications of
San Francisco, California

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704-0778
(510) 843-6200

PATENT

Attorney Docket No. GCENP006

TECHNIQUES FOR PROVIDING INTEROPERABILITY AS A SERVICE

RELATED APPLICATION DATA

[0001] The present application is related to U.S. Patent Application No. 09/820,964 for SYSTEM AND METHOD FOR MAPPING OF SERVICES filed March 30, 2001, U.S. Patent Application No. 09/820,965 for SYSTEM AND METHOD FOR INVOCATION OF SERVICES filed March 30, 2001, U.S. Patent Application No. 09/820,966 for SYSTEM AND METHOD FOR ROUTING MESSAGES BETWEEN APPLICATIONS filed March 30, 2001, U.S. Patent Application No. 10/727,089 for APPARATUS AND METHODS FOR PROVISIONING SERVICES filed December 2, 2003, U.S. Patent Application No. 10/728,356 for APPARATUS AND METHODS FOR CORRELATING MESSAGES SENT BETWEEN SERVICES filed December 3, 2003, and U.S. Patent Application No. 10/742,513 for APPARATUS AND METHODS FOR MEDIATING MESSAGES filed December 19, 2003, the entire disclosures of all of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to techniques for providing interoperability between and among disparate applications and services. More specifically, embodiments of the invention provide and manage highly individualized, on-demand access to applications and services in a network environment.

[0003] Corporate reliance on technology has become more complex and pervasive. Increasingly, companies are identifying opportunities to extend their core business or cut costs using the Internet. Both trends have put increasing priority on integrating the operation of disparate business applications that exist in different enterprises. As a result, the enterprise application integration (EAI) and business-to-business B2B industries have emerged to provide solutions for unifying enterprise legacy systems that may span corporate boundaries and may include the applications of business partners and customers. Ideally, this unification does not require sweeping changes to the underlying applications and data structures.

[0004] EAI and B2B solution providers typically offer end point solutions for managing business process interactions between end points. This can take place within an enterprise on a local network or, in the case of B2B, across the Internet. Although a specific enterprise software package may be designed to transparently handle diverse business processes carried out by two or more end nodes, each specific enterprise software package requires releasing, implementing or building customized connectors or adapters to connect to different legacy systems which will work for the specific business processes and applications used by the specific end nodes. As a result, these enterprise solutions are not easily scalable. Additionally, scores of adapters are needed for each vendor (e.g., Oracle, SAP and Peoplesoft). As each supplier releases new versions of their software, EAI and B2B vendors find themselves unable to gain traction under the burden of supporting existing adapters.

[0005] Notwithstanding the benefits of EAI and B2B solutions, the software costs and resource investments required often prevent small-to-medium enterprise (SME) customers from embracing EAI and B2B solutions. For SMEs, reliance on web services technology providers represents an increasingly attractive alternative.

[0006] The application service provider (ASP) market is one of the fastest growing segments of the software industry. ASPs make enterprise applications (e.g., human resources administration, recruiting, travel and expense management, sales force automation) available to customers over the web on a subscription basis. These applications are fully managed and hosted by the provider providing significant cost savings to enterprises and eliminating many of the issues requiring EAI solutions.

[0007] Some ASPs merely host and manage third-party packaged software for their customers (i.e., "managed hosters"). Others build new applications from the ground up to take advantage of the benefits and cost-savings of the ASP model. ASPs enjoy the profit margins and operational scalability of consumer Web companies like eBay and Yahoo, while at the same time offering the feature sets of complex enterprise software applications such as PeopleSoft and Siebel.

[0008] Although the ASP approach allows a business and its partners to use third party or custom applications, this approach does not allow the configuring and dismantling of complex arrangements between business partners. Specifically, the ASP approach requires custom configurations when business partners use different data formats for their messages or different communications protocols. Using these custom configurations, business partners specify the format of outgoing messages to comport with the recipient's format requirements. These messages can then be delivered to a recipient in a format understandable to the recipient. According to this approach, business entities must keep track of formatting and integration requirements of each of their recipient business partners in order to achieve interoperability. This can be costly and time-consuming.

[0009] None of these ad hoc approaches to interoperability can practically provide a single solution for facilitating the consumption of the wide array of disparate services employed by the typical enterprise. Moreover, none of these approaches is well suited to deliver such an array of services in the personalized manner to which so many users of the World Wide Web have become accustomed.

[0010] In view of the above, there is a need for facilitating communications between and among diverse business entities, processes, and services in a scalable manner. In addition, there is a need for techniques for mediating such communications in a manner which individualizes each user's experience.

SUMMARY OF THE INVENTION

[0011] According to the present invention, techniques are disclosed for providing interoperability between and among disparate platforms, services, or applications as a service. Various embodiments of the invention provide for an interoperability system or network which is operable to facilitate communication among disparate entities including, for example, individual users from one or more enterprises and a wide variety of enterprise applications and services from different service providers. Interoperability networks implemented according to such embodiments provide the mechanisms by which such entities may interact substantially transparently, as well as a policy-based directory capability to facilitate management of the manner in which such entities and their various applications are allowed to interact. As will become clear, the interoperability networks of the invention are capable of interacting with a virtually unlimited variety of computing models and platforms, thus ensuring the efficient and reliable delivery of services irrespective of the manner in which computing and networking technology evolves in the future. In addition, various embodiments of the invention are operable to facilitate the delivery of such services in a highly individualized, on-demand manner.

[0012] According to a specific embodiment of the invention, an interoperability system is operable to provide access to a plurality of services by a plurality of users. Each of the plurality of users is associated with one of a plurality of independent enterprises, and the plurality of services are associated with and controlled by a plurality of independent service providers. The plurality of services employ a plurality of interfaces at least some of which are not directly interoperable. At least one data store has a directory stored therein which maps an identity corresponding to each of the users to a policy framework which defines access policies relating to the services. The identity for each user identifies the associated enterprise. The at least one data store also has a plurality of rich client objects stored therein which are operable to be launched within browser environments on the client machines and to interact with the services via the interoperability system. At least one computing device is operable to connect with each of the client machines and each of the interfaces associated with the services, to selectively upload the rich client objects to the client machines with reference to the directory, and to selectively facilitate interaction among the uploaded rich client objects and the services with reference to the directory and the policy framework, thereby enabling the users associated with different ones of the enterprises to independently access the plurality of services using the interoperability system.

[0013] According to various specific embodiments, selected ones of the rich client objects are operable to interact with each other on the client machines, thus facilitating the delivery of front end services. The rich client objects may be uploaded during an initial sign-on process initiated by a sign-on request. Rich client objects may also be uploaded in response to a subsequent request for a service subsequent to the sign-on process. Previously installed rich client objects on the client machines may also be employed.

[0014] According to various specific embodiments, the interoperability system of the present invention may connect with client machines directly via a public wide area network. Alternatively, the system may connect with client machines via an enterprise network.

[0015] According to some embodiments, the interoperability system can facilitate "occasionally connected computing" by allow client machines to retain uploaded rich client objects and other uploaded data when the client machines are not connected to the system. According to one such embodiment, offline data are generated and cached by the rich client objects when the client machines are not connected to the system. The offline data are then received by the system when the client machines reconnect to the system. According to another embodiment, the system generates and caches offline data when the client machines are not connected to the system, and then transmits the offline data to the client machines when they reconnect to the system.

[0016] According to further embodiments, the interoperability system is operable to facilitate interaction between two or more of the services, thereby providing access to a composite service.

[0017] According to another embodiment, a computer-implemented method provides access to a plurality of services by a plurality of users having associated client machines. Each of the plurality of users is associated with one of a plurality of independent enterprises. The plurality of services is associated with and controlled by a plurality of independent service providers. The services employ a plurality of interfaces at least some of which are not directly interoperable. Rich client objects are selectively transmitted to the client machines. The rich client objects are operable to be launched within browser environments on the client machines and to interact with the services in accordance with a directory which maps an identity corresponding to each of the users to a policy framework which defines access policies relating to the services. The identity for each user identifies the associated enterprise. Interaction among the transmitted rich client objects and the services is selectively facilitated in accordance with the directory and the policy framework, thereby enabling the users associated with different ones of the enterprises to independently access the plurality of services using a single system.

[0018] According to another embodiment, an interoperability system and associated methods are provided for providing access to a plurality of services by a plurality of users having associated client machines. Each of the plurality of users is associated with one of a plurality of independent enterprises. The plurality of services are associated with and controlled by a plurality of independent service providers and employ a plurality of interfaces at least some of which are not directly interoperable. The system includes at least one data store having a directory stored therein which maps an identity corresponding to each of the users to a policy framework which defines access policies relating to the services. The identity for each user identifies the associated enterprise and a role associated with the user in the associated enterprise. The system also includes at least one computing device which is operable to connect with each of the client machines and each of the interfaces associated with the services. The at least one computing device selectively facilitates interaction among the client machines and the services with reference to the directory and the policy framework. The at least one computing device also facilitate consumption of the services in a unique manner for each user in accordance with the corresponding identity.

[0019] A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Fig. 1A illustrates the provisioning of a service for use over a network in accordance with one embodiment of the present invention.

[0021] Fig. 1B illustrates the mediation of messages sent from a first service to a second service via an interoperability network in accordance with one embodiment of the present invention.

[0022] Fig. 2 is a flowchart illustrating a procedure for mediating messages between services in accordance with one embodiment of the present invention.

[0023] Fig. 3 is a flowchart illustrating a procedure for mediating a message sent to a particular service in accordance with one embodiment of the present invention.

[0024] Fig. 4 is diagrammatic representation of a translation of a message from a SOAP format to an AS2 format in accordance with one embodiment of the present invention.

[0025] Fig. 5 is diagrammatic representation of a translation of a message from a MIME format to a DIME format in accordance with one embodiment of the present invention.

[0026] Fig. 6 is diagrammatic representation of a process for translating a message in accordance with one embodiment of the present invention.

[0027] Fig. 7 is a simplified diagram of a network environment in which specific embodiments of the invention may be implemented.

[0028] Fig. 8 is a diagrammatic representation of an exemplary computing device which may be employed to implement aspects of various embodiments of the present invention.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

Reference will now be made in detail to specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims. In the following description, specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In addition, well known features or details may not have been described to avoid unnecessarily obscuring the invention.

[0029] According to various embodiments of the invention, an interoperability network is provided which facilitates interoperability using, among other things, a wide variety of web services technologies and standards including, for example, SOAP, Web Services Description Language (WSDL), WS-Security, WS-Policy, and Business Process Execution Language (BPEL). The interoperability network mediates the technology differences in data formats, communications protocols and business policies through a set of established and defined policies.

[0030] In general, the term web services refers to a collection of technology standards which enable software applications of all types to communicate over a network. A web service typically facilitates a connection between two applications or services in which queries and responses are exchanged in XML over HTTP (or HTTPS). More specifically, the term web services implies the implementation of a stack of specific, complementary standards.

[0031] Although not specifically tied to any transport protocol, web services build on Internet connectivity and infrastructure to ensure nearly universal reach and support. In particular, web services take advantage of HTTP, the same connection protocol used by Web servers and browsers. XML (and it's corresponding semantics) is a widely accepted format for exchanging data. It is a fundamental building block for nearly every other layer in the web services stack. The Simple Object Access Protocol (SOAP) is a protocol for messaging between applications. It is based on XML and uses common Internet transport protocols like HTTP to carry its data. Web Services Description Language (WSDL) is an XML-based description of how to connect to and communicate with a particular web service. A WSDL description abstracts a particular service's various connection and messaging protocols into a high-level bundle and forms a key element of the UDDI directory's service discovery model. Finally, Universal Description, Discovery, and Integration (UDDI) represents a set of protocols and a public directory for the registration and real-time lookup of web services and other business processes. Various embodiments of the invention employ these and similar technologies to provide interoperability between and among disparate platforms, services or applications as a service.

[0032] Specific mechanisms by which interoperability networks implemented according to various embodiments of the invention may facilitate interaction among a variety of entities will now be described with reference to the accompanying figures. It will be understood that the mechanisms described are merely examples of techniques which may be employed to facilitate the basic functionalities of such interoperability networks. That is, any technologies which facilitate "on-demand" access to a wide range of services are within the scope of the invention.

[0033] Fig. 1A illustrates the provisioning of a service via a network in accordance with one embodiment of the present invention. As shown, the network includes an interoperability network 106 (which may be a portion of an interoperability network designed according to the invention) for facilitating the provisioning of services for use by entities having access to the network. In a specific example, a service (or set of services) 104 is provisioned by a provider 102 in conjunction with interoperability network 106. During the provisioning process, format connection, and security preferences may be specified for messages received by the service 104 as further described below. In one embodiment, provisioning includes setting up a service configuration such that the service may be used in the network. As part of this set up, the service can specify the type of message format that it prefers to receive. In the illustration, service(s) 104 is shown within provider 102, e.g., inside the enterprise firewall. However, it should be understood that service(s) 104 may reside outside the firewall at, for example, a third party site.

[0034] In some embodiments, provider 102 may optionally specify which users or services may access the provisioned service 104 and the conditions under which they may have access. It should be recognized that the service 104 can be provided by provider 102 to any type of entity such as, for example, an individual user from a particular organization or a particular organizational entity. An organization may represent a distinct business entity, a particular user of a business entity, or an administrative domain of a computer application.

[0035] As used herein, the term "service" may represent any computer application, process, entity, or device accessible to other applications, processes, entities, or devices through an interface such as an application programming interface (API), user interface, or Internet web user interface by any of a variety of protocols over a network within an entity or over the Internet. A service may also comprise multiple methods or applications on a single device or distributed across multiple devices.

[0036] Although not shown in Fig. 1A, provider 102 may provision any number and type of services. Also, any number and type of providers may provision services to be accessed through the interoperability network 106. Accordingly, the interoperability network 106 may be configured to provision multiple services from multiple providers. Additional mechanisms and techniques for provisioning services and for enabling disparate entities to interact according to the invention are described in U.S. Patent Applications No. 09/820,964, 09/820,966, 09/820,965, 10/727,089, 10/728,356 and 10/742,513 incorporated herein by reference above. Any of the mechanisms described in these referenced applications may easily be applied with the techniques described herein.

[0037] After services are provisioned, messages may then be sent between two or more services via the interoperability network. That is, a particular service may be accessed by another service via the network. For example, a user associated with a first device may access a particular service on a second device via the interoperability network using a communication process (or service) located on the first device.

[0038] Fig. 1B illustrates the mediation of messages sent from a first service to a second service via an interoperability network in accordance with a particular embodiment of the present invention. As shown, a message is being sent from service 110 to service 104 through interoperability network 106 which is accessible over a wide area network such as, for example, the Internet. Such a message may correspond to a request from a user associated with service 110 for access to service 104 which resides on a remote device. The request may be sent to the service 104 by a web application (*e.g.*, service 110) located on another remote device. In particular cases, the services may be configured to execute on their own and a user is not required to send a request or message to a particular service.

[0039] In one configuration of the present invention, interoperability network 106 includes any number of mechanisms for mediating communications between two or more services. In the illustrated embodiment, interoperability network 106 includes a mechanism for translating messages sent between the services, such as from service 110 to service 104. Messages can use formats such as MIME, DIME, and the like, with AS2, SOAP, and other application bindings. MIME and DIME are attachment/part formats, while SOAP and AS2 are application logic binding protocols. Of course, a message may use any suitable type of protocol, structuring, or formatting specification which results in a particular format for the message. When different entities use different formats for their messages, the interoperability network translate the messages such that recipients receive the messages in the appropriate format.

[0040] In an exemplary embodiment, a message having a MIME format is sent by service 110 and received into interoperability network 106 via path 116. Of course, the routing path 116 may include any number and type of routers and/or processing nodes. Interoperability network 106 then determines, e.g., through policies in the directory, that service 104 expects messages to be received in a DIME format and translates the message from MIME to DIME along path 114. The path 114 may include any number and type of routing devices (or services) and/or processing device (or services). The translated message, which is now in DIME format, is then sent to service 104 via path 112 which may include any suitable number and type of routing devices and/or processing nodes.

[0041] In addition to this transformation, any number of other enrichments may be applied to messages in the network. Such enrichments may include, for example, a digital signature service, a tariff calculator for a purchase order, etc.

[0042] According to various implementations, service 110 and the provider of service 110 need not be aware of the message format requirements of the message destination (service 104), nor of any format translation taking place in the interoperability network. Service 110 may send the message as if service 104 employed the same message format as used by service 110. A more detailed discussion of exemplary processes for mediating messages in different formats is provided below with regard to Figs. 2 through 6.

[0043] In addition to providing mechanisms for provisioning services and mediating messages sent to such services, interoperability network 106 also preferably includes a repository or a directory for storing various information regarding the services and entities which provision and/or use such services. This information may include, for example, user identities, service identities, and policies which control which entities in the network can interact, and the manner in which they can interact. The interoperability network preferably also includes mechanisms for creating and combining services, registering users and their identifying information, and handling messages routed between services and/or users. The repository may be formed from one or more databases or directory services, including LDAP, or the like stored on one or more memory devices on one or more computing platforms.

[0044] According to specific embodiments of the invention, the interoperability network provides security management including authentication, authorization and security policy enforcement using the information in the directory and policy framework. The network may perform security management at various points in a message's network lifecycle, e.g., when a message is sent into the network from a service, when it is routed to its destination endpoint, and when the message is delivered out of the network to its destination service. While the following discussion employs the term "service," it will be understood that this is intended to include all software entities capable of connecting to and interacting with the interoperability network.

[0045] Authentication is the process of verifying that users or services interacting via the network have valid network identities. The authentication process may also involve the network supplying credentials required by the service to identify the network. Authorization is the process of making sure a service has permission to exchange messages with another service. Security policy enforcement allows services to specify the level of security other services must employ to interact with them via the network. For example, if a first service has a security policy of required encryption for data and required password authorization or better, then only services connecting to the network with a connection security policy that requires at least data encryption will be allowed to exchange messages with the first service. Providers of services can define equivalent security policies, allowing the network to consider certain policies to be equivalent to others, though they are not the same, for the purpose of gaining access to services.

[0046] According to a specific embodiment, a service identity is the network service address of the network endpoint the connecting service wants to be associated with. The service proof is the password configured for the associated endpoint. The user identity is a combination of organization and user name. Optionally a service name may be provided with the user identity. The user identity associates the connecting service with the corresponding network user account.

[0047] When a service posts a message to the network or polls for a message, the service initiates a connection to the interoperability network. The service is authenticated and associated with an endpoint on the network. The network verifies that the connection security policy of the connecting service is at least as high as the connection security policy defined by the associated endpoint. If the authentication and security policy checks pass for a posted message, the message is accepted into the network and is ready to be routed. Otherwise the message is not accepted and a SOAP fault is returned to the service. If the service is polling for a message and verification succeeds, the message requested by the poll is delivered to the service. If security verification fails, a SOAP fault is returned to the service and the polled for message is not delivered.

[0048] When connecting to the network the service supplies a username that identifies the service as an identity on the network. The provided identity associates the connecting service with an endpoint service on the network. In addition, the service supplies a password and/or a client certificate as proof of that identity. In the case of HTTPS connections, the network provides a server certificate that may be used by the service for authentication of the network.

[0049] As mentioned above, each service must connect with a security level that is the same or higher than the connection security policy configured for the service's associated endpoint service which may be configured, for example, for HTTP, HTTPS (HTTP with encryption) or HTTPS with certificate-based authentication.

[0050] The network determines the endpoint associated with a message and routes the message to a message queue associated with that endpoint. During this routing phase, security policy and permission verification is performed. If the security policy and permission verification passes, the message is routed to the message queue associated with the destination endpoint. If either part of the verification does not pass, the message is not routed and a SOAP fault is returned to the service that originated the message.

[0051] Security policies are enforced in a bi-directional manner. That is the security policy of the connecting service's endpoint (i.e., the origin endpoint) and the security policy of the destination service's endpoint (i.e., the destination endpoint) must both be met. For example, if the origin endpoint has a security policy of HTTP, it will allow services that use HTTP or HTTPS to connect. However, the only endpoints the origin endpoint will be allowed to message with are endpoints with a security policy that allows HTTP. That is, endpoints with a security policy of HTTPS or higher will not allow services that connect with HTTP to message with the service associated with them.

[0052] Permission enforcement may also performed during the message routing phase. The destination endpoint has a permissions or access control list policy that is stored in the directory that the network references to determine whether or not the origin endpoint is allowed to exchange messages with this destination endpoint.

[0053] When a message is pushed by the network to a destination service, the network may perform authentication of the service, may provide authentication credentials to the service, and will enforce the connection security policy configured for the endpoint corresponding to the destination service. If authentication verification and security policy validation succeed, the message is delivered to the destination service. If either security verifications fail, the message is not delivered and a SOAP fault may be returned to the service that originated the message.

[0054] When connecting from the network to a destination service, the network may be configured to provide no authentication credentials, to supply a username and/or password, or to authenticate a digital certificate (e.g., a Verisign X.509 certificate) sent by the destination service to the network. In addition, the network may be configured to supply a digital certificate which the destination service can use to authenticate the network.

[0055] It will be understood that the foregoing description relating to security management is merely exemplary and that any suitable alternatives for providing any combination of the described functionalities are within the scope of the invention.

[0056] Fig. 2 is a flow process diagram illustrating one example of a technique for mediating messages sent to a particular service through an interoperability network implemented according to the invention. According to various embodiments, an interoperability network allows communication between services that may use different data formats or protocols. At 202, a format preference for a particular service is received. The format preference may include message format information indicating how messages should be received by that particular service. According to various embodiments, a message may incorporate any of the following format types: Direct Internet Message Encapsulation (DIME), Multipurpose Internet Mail Encapsulation (MIME), Simple Object Access Protocol (SOAP), or AS2, as well as any other protocols or format types.

[0057] The format preferences for the first service may be received in a variety of ways. In one example, format preference information is received over a network from a provider provisioning a service with the interoperability network. In the example of Fig. 1A, when provider 102 provisions service 104 with interoperability network 106, provider 102 specifies a format preference for messages received by service 104. According to various embodiments, format information is received when a provider or a service is introduced into the interoperability network. In another embodiment, format information is received when a provider or service is configured for operation with an interoperability network. In a preferred embodiment, a provider specifies only a format preference. The provider does not need to perform any other actions, such as configuring transformation mappings, specifying translation processes, setting up routing to format conversion services, etc.

[0058] At 206, the format preference information for the particular service is stored and associated with such particular service. Information can be stored in a variety of ways or in any number and type of storage devices, such as one or more databases, caches, directory services, or any of a variety of data structures. In one implementation, format preference information is stored in an interoperability network database. In another implementation, format preference information is stored at another network entity associated with the interoperability network. In a specific implementation, a format table is used to store entries regarding each service and its associated format preference.

[0059] At 208, messages sent to a provisioned service are mediated using the format preference stored for that service. Details of this mediation according to specific embodiments of the invention will now be described with reference to Figs. 3 through 6.

[0060] Fig. 3 is a flow process diagram showing one example of a technique for mediating messages sent to a particular service through an interoperability network (e.g., network 106 of Fig. 1). According to various embodiments, messages sent to the particular service may be received into the interoperability network from a variety of different entities including other services, applications, hosts, network storage arrays, or terminals. The messages are sent to the interoperability network over a wide area network such as the Internet.

[0061] At 302, the interoperability network receives a message addressed or requesting access to a particular service. In the example of Fig. 1, service 110 sends such a message to the network requesting service 104. According to various embodiments, the message addressed to a particular service may include the particular address associated with the particular service. In other embodiments, the message merely includes some indicator associating the message with the desired service or the provider of the desired service.

[0062] At 304, the format of the message is determined. In one example, determining the format may involve scanning the message to determine if formatting mechanisms such as DIME, MIME, SOAP, or AS2 are used. In some examples, multiple formatting mechanisms may be used. In an alternative embodiment, the format of the message may be presumed to be the same as the format preference specified for the sending service. However, a verification of the format is preferably performed by analyzing each message.

[0063] At 306, the format of the message is compared to the stored format preference for the requested service (i.e., the receiving service). For example, if the format preference stored for the particular service indicates that the particular service should receive messages with DIME formatting and the message format is MIME, translation may be needed. If it is determined that translation is necessary at 308, the message is translated at 310 and sent to the particular service at 312. If translation is not necessary at 308, the message is forwarded to the particular service at 312.

[0064] According to various embodiments, the techniques of the present invention recognize whether translation is necessary without a received message explicitly indicating that a translation is needed. Traditional mechanisms typically involve express requests for translation. In contrast, according to various embodiments of the present invention, a provider of a particular service need only specify the format of messages that the particular service is configured to receive. That is, the formatting used by other services (even services which access the particular service) need not be specified. Techniques of the present invention allow transparent translation of formats without involving user intervention.

[0065] Fig. 4 is diagrammatic representation of a translation of a message from a SOAP format to an AS2 format in accordance with one embodiment of the present invention. The spacing shown for each message type is merely for clarity purposes and is not meant to accurately represent the formatting of actual SOAP and AS2 type messages. As shown, SOAP message 400 includes headers that provide various information such as the format of the message. In particular, content type 402 indicates that the message is multipart/related. Although not shown, other information can be provided in the headers as well. Next, SOAP message 400 includes a MIME boundary 404. Between MIME boundary 404 and 418, a SOAP envelope is included. The content type 406 of this bounded content is indicated as text/XML. The SOAP envelope contains a SOAP header 410 and a SOAP body 412. The content of the SOAP envelope as shown is exemplary only and does not reflect the actual variations and content possible for SOAP envelopes.

[0066] The SOAP message 400 further contains a second part between MIME boundaries 418 and 424. The content type 420 of these contents is indicated as application/edi-x12. The content 422 includes an X.12 EDI document. Other MIME headers beyond content type may be present as well. MIME boundary 424 marks the conclusion of these contents and can mark the beginning of another set of contents. Any number of sets of contents separated by MIME boundaries can be included in such a SOAP message 400. Other content types may include other types of text file formats, image file formats, video file formats, audio file formats, executable formats, etc. Alternatively, MIME boundary 424 can be a terminal boundary marking the end of the message.

[0067] In the present embodiment, when SOAP message 400 is translated to AS2 message 440, SOAP part 416 is stripped from the message and headers 402 from SOAP message 400 are replaced with AS2 headers 426. In addition, a portion of SOAP message 400 between MIME boundary 418 and 424 is included in AS2 message 440 as follows: content type 420 in SOAP message 400 is included as content type 428 in AS2 message 440, and content 422 in SOAP message 400 is included as content 432 in AS2 message 440.

[0068] Various MIME headers associated with the MIME part would also be included in the translated message. If SOAP message 400 contains additional contents (MIME parts) beyond boundary 424, those contents could also be transferred to the AS2 message, in which case the multiple parts of the AS2 message would be separated by MIME boundaries. Alternatively, each of the separate additional MIME parts of the SOAP message could be transformed into additional separate AS2 messages. AS2 message 440 may furthermore contain an added digital signature, in which case the message would contain multiple parts separated by MIME boundaries.

[0069] In the present exemplary embodiment, SOAP message 400 is translated to AS2 message 440. It should be recognized that an AS2 message can also be translated to a SOAP message in other embodiments. In such an example, a SOAP part is added to the AS2 message so as to result in a SOAP message. The AS2 headers are replaced by headers appropriate to a SOAP message. The content type and content of the AS2 message is inserted as a MIME part following the SOAP MIME part in the resulting SOAP message. Any suitable number of AS2 message parts may also be combined into a single SOAP message with multiple MIME boundaries. Similarly, any message format can be translated to another message format in accordance with various embodiments of the present invention.

[0070] Fig. 5 is diagrammatic representation of a translation of a message from a MIME format to a DIME format in accordance with another embodiment of the present invention. The spacing shown for each message type is merely for clarity purposes and is not meant to accurately represent the formatting of actual MIME and DIME type messages. As shown, MIME message 500 includes headers 504 and MIME boundaries 506 and 510. Content 508 is located between MIME boundaries 506 and 510. Additional sets of contents and MIME boundaries can be included after MIME boundary 510 according to various embodiments.

[0071] In one such embodiment, when the MIME message 500 is translated to DIME message 502, headers 504 from MIME message 500 are translated to headers 512 in DIME message 502. In particular, a content type header in MIME message 500 may have a value of "multipart/related" but will have a value of "application/dime" in DIME message 502. Furthermore, the content type and content id headers 507 in the first MIME part of MIME message 500 are stored inside DIME record descriptor 513 inside DIME message 502.

Content 508 of the MIME part is included as DIME record data 514. DIME record descriptor 513 includes information about the length of content 514, so there is no need to include boundaries in a DIME message such as MIME boundaries to determine the beginning and end of a set of contents.

[0072] MIME message 500 is parsed to determine the location of MIME boundaries 506 and 510. Once these boundaries are found, content 508 can be included as content 514, and information about the length of content 514, based on the distance between the boundaries or amount of information between the boundaries, can be included in record descriptor 513. Although the present embodiment includes content type and content ID in MIME headers 507, it should be recognized that additional header information can also be included. Such additional header information may also be stored in the DIME message.

[0073] Although the present exemplary embodiment includes translating MIME message 500 to DIME message 502, it should be recognized that a DIME message can also be translated into a MIME message. For example, information about the length of record data 514 that is stored in record descriptor 513 can be used to determine the placement of content 508 and MIME boundaries 506 and 510 in a MIME message 500. Similarly, any other message format can be translated to another message format according to various embodiments of the present invention.

[0074] Fig. 6 is a flow process diagram depicting one example of a technique for translating between formats. At 601, the message format of the received message is determined. According to various embodiments, the message format can be MIME, DIME, other, or the message can have no attachment formatting. In one example, the format is DIME if the content type is "application/dime" and MIME if the content type is "multipart/*" (where * can be a number of different types). Otherwise, the format may be neither. At 603, the desired format used by the destination service is determined. According to various embodiments, it may be determined that the desired format is the same as the format of the received message at 605. In this case, no translation is necessary, except that some headers or other SOAP parts may be altered and/or replaced at 607 as necessary to reflect an outgoing message from the interoperability network versus a message received into the network. At 609, the remaining message contents can be copied into the outgoing message in an unmodified manner without need for processing.

[0075] If it is determined that translation from MIME to DIME is needed at 611, the content type of the message is set to "application/dime." At 613, the main SOAP portion of the message is written as a record with desired content type information. At 615, attachments to the original message are parsed. For each attachment, a new DIME record copying only some MIME headers are written. In one example, Content-ID fields are unchanged when copied to a DIME format and Content-Type fields are also unchanged when copied to a DIME format. The type format field in the DIME record is set to MIME media type.

[0076] If the Content-Transfer-Encoding header is set for a MIME part, the content of the MIME part is decoded since DIME does not support transfer encoded record content. The content is then written in the decoded format. Next, if appropriate, the record can be chunked. DIME generally allows for a single part to be written as multiple records. Since DIME records declare their size in the beginning of the record, chunking allows for more optimal transmission of large amounts of data and streaming.

[0077] At 621, if it is determined that translation from DIME formatting to MIME formatting is needed, the content type for the new MIME message is set to "multipart/related," with proper boundary attribute, at 623. Next, a boundary for the MIME message to be composed is generated at 625, in order to separate the new MIME parts. For each portion of the message, boundaries, headers, and payloads are written at 627. In particular, each DIME record is parsed and a new MIME part is written for each record. This process involves 1) writing a MIME boundary, 2) writing MIME headers, 3) writing data content, and 4) writing another MIME boundary.

[0078] More specifically, once the first MIME boundary is written, the MIME headers are written. If the content-type is a MIME type, the content-type can be copied as-is. If content type is URI, then the content-type can be set to "text/xml." If the content-type is in some other form, then the content-type can be set to "application/octet-stream." In addition, if a content-ID is present, it can be left unchanged in the new message. If the record is not chunked, the content-length can be set to the size of the data record. Otherwise, if the record is chunked, then the content-length header may not be set.

[0079] Next, the data content of the record is written. In one example, if the data record is the first record (i.e., SOAP part), then the SOAP part may be parsed to replace SOAP headers used by the interoperability network. After the data content of the record is written, a second MIME boundary is written. If the data record is the last record, the MIME boundary is set as the closing boundary.

Analogous threads for converting AS2 to SOAP (629-635) and SOAP to AS2 (637-643) may also be included.

[0080] In the embodiment shown, if it is determined that formatting is not DIME, MIME, AS2, or SOAP at 645, the message headers are copied unchanged at 647. According to various embodiments, the SOAP content is also copied, but the SOAP content is parsed in order to replace SOAP headers used by the interoperability network.

[0081] It should be noted that Fig. 6 describes one example of translation mechanisms for very specific formats. A variety of other formats and translation mechanisms are also available. For example, various embodiments of the invention support communication and interoperability with inbound FTP, secure FTP, HTTP, secure HTTP, all web services standards, and a wide variety of legacy applications. Embodiments of the invention also provide a wide variety of connectors to, for example, web portals, application services, desktop applications, handheld devices, and any software designed for web services.

[0082] As discussed above, the vast majority of application services today are provided according to a computing model which is characterized by several limitations. According to this model, a user wishing to access any of a number of application services employs a client machine (e.g., a desktop computer) to generate a request for a particular service. This is typically facilitated by a browser operating on client machine. The browser is an application which communicates via a network (e.g., a public or private LAN or WAN) with a server which manages access to the application services. What is typically viewed by the user is a page (e.g., an HTML page) which is generated by the server and delivered over the network for display in the user's browser.

[0083] The server in this model typically employs a three-tiered architecture to manage access to the associated application services. A portal layer governs display of information presented in the client's browser. An application server layer manages access to the application services on a high level, but because of the varied nature of the application services, an integration layer is required to normalize the communications with these services. That is, the integration layer, which is the primary focus of EAI providers, facilitates connection with and communication among the application services themselves. Unfortunately, the great variety of application services and the highly individualized nature of EAI solutions coupled with the limitations of this computing model have made EAI an economically impracticable approach for all but the largest enterprises.

[0084] By contrast, embodiments of the invention described above with reference to Figs. 1 through 6 address many of the limitations associated with the paradigm currently dominating the EAI and B2B integration industries by abstracting the integration and interoperability functionalities from the conventional computing model and, instead, providing interoperability between and among disparate platforms, services or applications as a service. As described above, this inventive approach provides an interoperability network having a directory capability which facilitates management of user identities (e.g., including role and group membership), service identities, and policies which control which entities in the network can interact, and the manner in which they can interact. By delivering services according to such an "on demand" model, EAI and B2B are brought within the means of small and medium-sized enterprises.

[0085] In addition, embodiments of the invention are contemplated in which each user's experience in the interoperability network is personalized. That is, in addition to mediating the technology and communication protocol issues between disparate platforms and enterprises, embodiments of the invention further leverage evolving computing paradigms in combination with the identity and policy management capabilities of the interoperability network to deliver services to each user in a highly individualized manner. Through the maintenance and management of rich metadata for each user, an interoperability network designed according to the invention may provide, as a service, a personal "portal" through which on-demand access to any service capable of connecting with the network may be provided.

[0086] According to various embodiments, many technologies developments may be leveraged to facilitate this unique user experience. For example, computing paradigms have moved increasingly toward the development of an array of software technologies commonly referred to as "rich clients." As used herein, a "rich client" refers to a software application or applet (or a collection of such software objects) which is operable to be launched in a browser on a client machine. A rich client typically includes both the display logic which governs what is displayed on the client machine, as well as some level of application logic (i.e., executable code) which provides functionality on the client machine which, in the past, has typically been provided by code on the remote server. Representative examples of rich client technology are provided by Dream Factory Software, Inc. of Los Gatos, California, General Interface Corporation of San Francisco, California, Macromedia Inc. of San Francisco, California, and Microsoft Corporation (e.g., .NET technology, Office suite, etc.).

[0087] In view of the foregoing, specific embodiments of the invention take advantage of the development of rich client technology to move further from conventional EAI and B2B paradigms and to provide access to application services in an even more flexible and efficient manner. A particular approach to implementing such an embodiment will now be described with reference to the simplified network diagram of Fig. 7.

[0088] As shown in the diagram, the browser on client machine 702 may communicate with interoperability network 704 without requiring an intermediate server. This communication may occur over an intervening network 706 using, for example, a modem (e.g., dial-up, DSL, cable, etc.). Embodiments of the invention are also contemplated in which an intermediate server, e.g., an enterprise's server, is employed to facilitate connection with network 704.

[0089] Interoperability network 704 (e.g., using one or more computing devices such as server 707) facilitates access to selected ones of associated services 708 which may be sponsored or provided by network 704, or may comprise application services from third parties. These services may actually reside in the network or be connected via intervening networks (e.g., 709). As mentioned above, network 704 provides transparent connections to and interoperability with a wide variety of services and applications. And like previously described embodiments, interoperability network 704 has a directory capability which facilitates management of user identities (e.g., including role and group membership), application service identities, and policies which control which entities in the network can interact, and the manner in which they can interact.

[0090] According to specific embodiments, the interoperability network employs the directory to manage interactions among the services associated with many independent organizations, each with different access, authentication and encryption technologies. Differences in organizational security policies are handled using a policy framework which mediates the differences. According to some embodiments, each organization is able to configure and enforce access rights with multiple methods of authentication being supported.

[0091] According to a specific embodiment, an interoperability network implemented according to the invention supports WS-Policy, a flexible mechanism which enables enterprises to govern access to the services they have deployed on the interoperability network. Such a mechanism may be employed, for example, to ensure that data are exchanged over encrypted connections to the interoperability network, that user and service identities are verified (using the directory), and that access to a particular service is limited and controlled. According to even more specific embodiments, such capabilities are supported using industry standards such as, for example, SSL, IPSEC VPNs, and X.509 digital certificates.

[0092] Referring back to Fig. 7, communication between interoperability network 704 and client machine 702 may be facilitated through the use of rich client technology. According to some of these embodiments, interoperability network 704 facilitates the transfer of rich client "widgets" 710 to client machine 702 in accordance with the access privileges corresponding to that client. As used herein, the term "widget" refers to one or more software objects which, when instantiated on a client machine, implements all or a portion of a rich client. One of the advantages associated with modern rich client technology is that they are all largely based on the same set of standards and protocols. Therefore, supporting the wide variety of available rich client technology presents a lesser technical challenge than the problem of integrating the widely disparate back end application services. As noted above, representative examples of rich client technology which may be employed with various embodiments of the invention are provided by Dream Factory Software, Inc. of Los Gatos, California, General Interface Corporation of San Francisco, California, Macromedia Inc. of San Francisco, California, and Microsoft Corporation (e.g., .NET technology, Office suite, etc.).

[0093] Interoperability network 704 either stores widgets 710 within the network (e.g., in data store 712), or has access to a store of widgets at some remote location(s). In any case, using the directory capabilities of the present invention, interoperability network 704 verifies the identity of the user and determines (with reference to that identity) which of the widgets are to be transferred to client machine 702. According to a specific embodiment, at least some of the widgets to which the client machine is entitled access are transferred to the client machine during an initial "single sign-on" session. Additional widgets may be transferred at run time as the user requests access to additional services. According to still more specific embodiments, some widgets or rich clients may already be installed and running on the client machine prior to the user signing on to the interoperability network. In any case, the widget or widgets dynamically instantiate within the client's browser and the resulting rich client(s) communicate with the specific services the user is entitled to use via interoperability network 704.

[0094] Notwithstanding the foregoing references to rich client technology, it should be understood that such technology is merely referred to herein for exemplary purposes and should not be considered to limit the scope of the invention. That is, any technologies (e.g., "thick" or "thin" clients) which can facilitate the highly individualized user experience described herein is within the scope of the invention. For example, templatized HTML pages delivered from the network according to a conventional client-server model may incorporate personalized views which achieve this end. In addition, client-side applications which do not fall within the conventional understanding of the term "rich client" may also be employed to personalize the consumption of services via an interoperability network designed according to the invention.

[0095] The approach described above with reference to Fig. 7 represents a significant shift from the approach in which the client machine interacts with the services network via an intermediate server (e.g., the application server on the enterprise's LAN). That is, according to these embodiments, the client machine interacts with the network in much the same way as any of the services or applications which interact with the network. And because the client machine represents an individual user, the delivery of services via the services network is fundamentally changed.

[0096] For example, using such an approach, in one browser window, the user may be interacting directly with a particular service, and in another with a complex process created within the services network involving multiple services. Because of the rich client capability on the client's machine which typically employ open standard APIs, the service in one window can interact with the process or services in the other. Thus, in addition to the back-end integration provided within the interoperability network (e.g., the composite services), embodiments of the present invention also enable sophisticated interaction on the user's machine (i.e., front end integration).

[0097] In a service oriented architecture, there is also a decoupling of the software development processes from a front consumption standpoint and the back end service definition in the system. That is, development of applications at the back end is traditionally a relatively slow moving process which relates to the development of application programming interfaces (e.g., services) for delivering a variety of desired functionalities. By contrast, the rate of innovation at the client side (which typically relates to how information is displayed) occurs much faster. And because it is the identity of the user, their role and group membership, which governs access to services, and because there is no requirement for an intervening portal, it is largely irrelevant what type of client technology is employed by the user's machine. That is, client technology (rich and otherwise) from any of a wide variety of providers (e.g., Dream Factory, General Interface, Macromedia, Microsoft, etc.) can be used to interact with the interoperability network of the present invention in a manner which is decoupled from the rate at which the application services evolve.

[0098] It should be understood that the embodiment of Fig. 7 could be employed in a context where client machine 702 is behind the firewall of an enterprise network which has its own systems in place for managing user identity, security, access, etc. In such an embodiment, the enterprise network would send a security assertion (e.g., in SAML) to the interoperability network for one of its users requesting access to services associated with the interoperability network. And based on the credentials being asserted, the individual's access to the widgets and services on the network will be determined.

[0099] However, it should also be understood that according to some embodiments, a company can implement a virtual enterprise network which requires only that its employees have some kind of computing device (e.g., personal computer, personal digital assistant, wireless device, etc.) and a connection to the Internet (e.g., wired or wireless modem, etc.). The remaining enterprise network infrastructure and functionality may be provided by interoperability network 704 through the use of rich client technology which is deployed from network 704. The advantages of such an approach for both large and small enterprises are manifest, allowing sophisticated enterprise network functionality to be enjoyed by even the smallest of enterprises. Simply put, according to the invention, an interoperability network designed according to the invention has the potential for allowing an enterprise to outsource much (if not all) of this functionality to facilitate consumption of a wide range of services in an "on demand" manner.

[00100] Another advantage associated with the use of rich client technology in conjunction with the integration service network of the present invention is the potential for dramatically reducing network traffic, and therefore scaling to a large number of users. That is, under conventional computing paradigms, the vast majority of traffic in the network involves the transmission of redundant information, e.g., display refreshes, rather than critical information, i.e., system-of-record transactions. By contrast, according to specific embodiments of the invention, once the rich clients are operational on the client machines, a very high percentage of the network traffic relates to system-of-record transactions because

things like display processing are being taken care of on the client machines themselves. This dramatic reduction in traffic corresponds to an equally dramatic potential for scaling the interoperability network.

[00101] Yet another advantage of the use of rich client technology in this context is that it facilitates so called "occasionally connected computing." Because of the high processing power of today's PCs and other mobile technology, and because of the capabilities of the many varieties of rich clients, a user can potentially continue to work even when not connected to the interoperability network. That is, he can connect with his enterprise via the interoperability network using any Internet connection, download any data and widgets required to access or take advantage of a given service, then disconnect and (assuming the stored policies corresponding to his identity allow him to do so) retain the widgets and/or data on his machine for working offline.

[00102] According to one such embodiment, information generated during such an offline work session is cached for transmission to the interoperability network (or the enterprise via the interoperability network) when the user logs back on. This cache or queue can be thought of conceptually as an "out box" on the user's machine which synchronizes with the network each time the user signs on to the network or affirmatively requests synchronization.

[00103] Similarly, an "in box" queue may be maintained in the interoperability network with information relevant to the user which is generated while the user is offline. For example, when an event occurs in the enterprise for which the user typically gets a notification, the notification can be stored in the user's in box until the user logs into the system or affirmatively requests the contents of the queue.

[00104] Referring now to Fig. 8, a computer system 800 suitable for implementing various aspects of the present invention (e.g., client 702 and/or server 707 of Fig. 7) is depicted. For example, one or more such computer systems may be employed in an interoperability network to manage the deployment of rich clients and/or the delivery of HTML to client computers. Computer system 800 includes one or more central processing units (CPUs) 802, one or more blocks of memory 804, input and output interfaces 806, and a bus 808 (e.g., a PCI bus). Alternatively, computer systems employing point-to-point infrastructures instead of buses may also be employed. When acting under the control of appropriate software or firmware, CPU 802 is responsible for implementing various portions of the techniques of the present invention. It preferably accomplishes all these functions under the control of software including an operating system and any appropriate applications software. CPU 802 may include one or more processors. In a specific embodiment, some portion of memory 804 (such as non-volatile RAM and/or ROM) also forms part of CPU 802. However, there are many different ways in which memory could be coupled to the system. Memory block 804 may be used for a variety of purposes such as, for example, caching and/or storing data, program code, etc.

[00105] The input and output interfaces 806 typically provide an interface to various I/O devices, such as mouse, keyboard, display, as well as providing a communication interface with other computer systems over a computer network. Among the communication interfaces that may be provided are Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, and the like. In addition, various very high-speed interfaces may be provided such as fast Ethernet interfaces, Gigabit Ethernet interfaces, ATM interfaces, HSSI interfaces, POS interfaces, FDDI interfaces and the like. Generally, these interfaces may include ports appropriate for communication with the

appropriate media. In some cases, they may also include an independent processor and, in some instances, volatile RAM.

[00106] It will be understood that the system shown in Fig. 8 is an exemplary computer system and is by no means the only system architecture on which the various aspects of the present invention can be implemented.

[00107] Regardless of system's configuration, it may employ one or more memories or memory modules (such as, for example, memory block 804) configured to store data, program instructions for the general-purpose network operations and/or the inventive techniques described herein. The program instructions may control the operation of an operating system and/or one or more applications, for example. The memory or memories may also be configured to store information in a repository directory.

[00108] Because such information and program instructions may be employed to implement the systems/methods described herein, the present invention also relates to machine readable media that include program instructions, state information, etc. for performing various operations described herein. Examples of machine-readable media include, but are not limited to, magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROM disks and DVDs; magneto-optical media such as floptical disks; and hardware devices that are specially configured to store and perform program instructions, such as read-only memory devices (ROM) and random access memory (RAM). The invention may also be embodied in a carrier wave traveling over an appropriate medium such as airwaves, optical lines, electric lines, etc. Examples of program instructions include both machine code, such as produced by a compiler, and files containing higher level code that may be executed by the computer using an interpreter.

[00109] While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention. For example, there are many different network topologies which may be employed to implement an interoperability network having some or all of the functionalities described herein. In addition, such a network may be deployed on the World Wide Web, some other portion of the Internet, or in some alternative network and remain within the scope of the invention. That is, any type of network or system which enables users and services from different enterprises and service providers to interact in the manner described is within the scope of the invention.

[00110] In addition, although embodiments have been described herein with reference to the use of rich client technology, it will be understood that this is merely a subset of the technologies which are operable to facilitate the basic functionalities of the present invention. That is, any technologies which can facilitate the highly individualized, on-demand consumption of services as described herein are within the scope of the invention.

[00111] Moreover, as described herein, the types of services which may be consumed using the various embodiments of the invention are quite diverse and are not limited by the references herein to particular examples of such services. For example, embodiments have been described herein which refer to web services and the various languages and protocols associated with web services. However, as noted above, the notion of a service is a much broader concept and should not be limited by such references.

[00112] In addition, although various advantages, aspects, and objects of the present invention have been discussed herein with reference to various embodiments, it will be understood that the scope of the invention should not be limited by reference to such

advantages, aspects, and objects. Rather, the scope of the invention should be determined with reference to the appended claims.